Rethinking the Form of Latent States in Image Captioning

Bo Dai[1](✉), Deming Ye[2], and Dahua Lin[1]

[1] CUHK-SenseTime Joint Lab, The Chinese University of Hong Kong,
Shatin, Hong Kong
{db014,dhlin}@ie.cuhk.edu.hk

[2] Department of Computer Science and Technology, Tsinghua University,
Beijing, China
ydm18@mails.tsinghua.edu.cn

Abstract. RNNs and their variants have been widely adopted for image captioning. In RNNs, the production of a caption is driven by a sequence of latent states. Existing captioning models usually represent latent states as vectors, taking this practice for granted. We rethink this choice and study an alternative formulation, namely using two-dimensional maps to encode latent states. This is motivated by the curiosity about a question: *how the spatial structures in the latent states affect the resultant captions?* Our study on MSCOCO and Flickr30k leads to two significant observations. First, the formulation with 2D states is generally more effective in captioning, consistently achieving higher performance with comparable parameter sizes. Second, 2D states preserve spatial locality. Taking advantage of this, we *visually* reveal the internal dynamics in the process of caption generation, as well as the connections between input visual domain and output linguistic domain.

1 Introduction

Image captioning, a task of generating short descriptions for given images, has received increasing attention in recent years. Latest works on this task [1–4] mostly adopt the encoder-decoder paradigm, where a recurrent neural network (RNN) or one of its variants, *e.g.* GRU [5] and LSTM [6], is used for generating the captions. Specifically, the RNN maintains a series of *latent states*. At each step, it takes the visual features together with the preceding word as input, updates the latent state, then estimates the conditional probability of the next word. Here, the latent states serve as pivots that connect between the visual and the linguistic domains.

Following the standard practice in language models [5,7], existing captioning models usually formulate the latent states as *vectors* and the connections between them as fully-connected transforms. Whereas this is a natural choice

B. Dai and D. Ye—Equal contribution.

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11209, pp. 294–310, 2018.
https://doi.org/10.1007/978-3-030-01228-1_18

for purely linguistic tasks, it becomes a question when the visual domain comes into play, *e.g.* in the task of image captioning.

Along with the rise of deep learning, convolutional neural networks (CNN) have become the dominant models for many computer vision tasks [8,9]. *Convolution* has a distinctive property, namely *spatial locality*, *i.e.* each output element corresponds to a local region in the input. This property allows the spatial structures to be maintained by the feature maps across layers. The significance of spatial locality for vision tasks have been repeatedly demonstrated in previous work [8,10–13].

Image captioning is a task that needs to bridge both the linguistic and the visual domains. Thus for this task, it is important to capture and preserve properties of the visual content in the latent states. This motivates us to explore an alternative formulation for image captioning, namely representing the latent states with 2D maps and connecting them via convolutions. As opposed to the standard formulation, this variant is capable of preserving spatial locality, and therefore it may strengthen the role of visual structures in the process of caption generation.

We compared both formulations, namely the standard one with vector states and the alternative one that uses 2D states, which we refer to as *RNN-2DS*. Our study shows: (1) The spatial structures significantly impact the captioning process. Editing the latent states, *e.g.* suppressing certain regions in the states, can lead to substantially different captions. (2) Preserving the spatial structures in the latent states is beneficial for captioning. On two public datasets, MSCOCO [14] and Flickr30k [15], RNN-2DS achieves notable performance gain consistently across different settings. In particular, a simple RNN-2DS without gating functions already outperforms more sophisticated networks with vector states, *e.g.* LSTM. Using 2D states in combination with more advanced cells, *e.g.* GRU, can further boost the performance. (3) Using 2D states makes the captioning process amenable to visual interpretation. Specifically, we take advantage of the spatial locality and develop a simple yet effective way to identify the connections between latent states and visual regions. This enables us to visualize the dynamics of the states as a caption is being generated, as well as the connections between the visual domain and the linguistic domain.

In summary, our contributions mainly lie in three aspects. First, we rethink the form of latent states in image captioning models, for which existing work simply follows the standard practice and adopts the vectorized representations. To our best knowledge, this is the first study that systematically explores two dimensional states in the context of image captioning. Second, our study challenges the prevalent practice, which reveals the significance of spatial locality in image captioning and suggests that the formulation with 2D states and convolution is more effective. Third, leveraging the spatial locality of the alternative formulation, we develop a simple method that can visualize the dynamics of the latent states in the decoding process.

2 Related Work

Image Captioning. Image captioning has been an active research topic in computer vision. Early techniques mainly rely on detection results. Kulkarni *et al.* [16] proposed to first detect visual concepts including objects and visual relationships [17], and then generate captions by filling sentence templates. Farhadi *et al.* [18] proposed to generate captions for a given image by retrieving from training captions based on detected concepts.

In recent years, the methods based on neural networks are gaining ground. Particularly, the encoder-decoder paradigm [1], which uses a CNN [19] to encode visual features and then uses an LSTM net [6] to decode them into a caption, was shown to outperform classical techniques and has been widely adopted. Along with this direction, many variants have been proposed [2,20–22], where Xu *et al.* [2] proposed to use a dynamic attention map to guide the decoding process. And Yao *et al.* [22] additionally incorporate visual attributes detected from the images, obtaining further improvement. While achieving significant progress, all these methods rely on *vectors* to encode visual features and to represent latent states.

Multi-dimensional RNN. Existing works that aim at extending RNN to more dimensions roughly fall into three categories:

(1) RNNs are applied on *multi-dimensional grids*, *e.g.* the 2D grid of pixels, via recurrent connections along different dimensions [23,24]. Such extensions have been used in image generation [25] and CAPTCHA recognition [26].
(2) Latent states of RNN cells are stacked across multiple steps to form feature maps. This formulation is usually used to capture temporal statistics, *e.g.* those in language processing [27,28] and audio processing [29]. For both categories above, the latent states are still represented by *1D vectors*. Hence, they are essentially different from this work.
(3) Latent states themselves are represented as multi-dimensional arrays. The RNN-2DS studied in this paper belongs to the third category, where latent states are represented as 2D feature maps. The idea of extending RNN with 2D states has been explored in various vision problems, such as rainfall prediction [30], super-resolution [11], instance segmentation [12], and action recognition [13]. It is worth noting that all these works focused on tackling visual tasks, where both the inputs and the outputs are in 2D forms. To our best knowledge, this is the first work that studies recurrent networks with 2D states in image captioning. A key contribution of this work is that it reveals the significance of 2D states in connecting the visual and the linguistic domains.

Interpretation. There are studies to analyze recurrent networks. Karpathy *et al.* [31] try to interpret the latent states of conventional LSTM models for natural language understanding. Similar studies have been conducted by Ding *et al.* [32] for neural machine translation. However, these studies focused on linguistic analysis, while our study tries to identify the connections between linguistic and visual domains by leveraging the spatial locality of the 2D states.

Our visualization method on 2D latent states also differs from the attention module [2] fundamentally, in both theory and implementation. (1) Attention is a *mechanism* specifically designed to guide the focus of a model, while the 2D states are a form of *representation*. (2) Attention is usually implemented as a sub-network. In our work, the 2D states by themselves do not introduce any attention mechanism. The visualization method is mainly for the purpose of interpretation, which helps us better understand the internal dynamics of the decoding process. To our best knowledge, this is accomplished for the first time for image captioning.

3 Formulations

To begin with, we review the encoder-decoder framework [1] which represents latent states as 1D vectors. Subsequently, we reformulate the latent states as multi-channel 2D feature maps for this framework. These formulations are the basis for our comparative study.

3.1 Encoder-Decoder for Image Captioning

The encoder-decoder framework generates a caption for a given image in two stages, namely *encoding* and *decoding*. Specifically, given an image I, it first encodes the image into a feature vector $\mathbf{v}$, with a *Convolutional Neural Network (CNN)*, such as VGGNet [19] or ResNet [8]. The feature vector $\mathbf{v}$ is then fed to a *Recurrent Neural Network (RNN)* and decoded into a sequence of words $(w_1, \dots, w_T)$. For decoding, the RNN implements a recurrent process driven by latent states, which generates the caption through multiple steps, each yielding a word. Specifically, it maintains a set of latent states, represented by a vector $\mathbf{h}_t$ that would be updated along the way. The computational procedure can be expressed by the formulas below:

$$\mathbf{h}_0 = \mathbf{0}, \quad \mathbf{h}_t = g(\mathbf{h}_{t-1}, \mathbf{x}_t, \mathbf{I}), \tag{1}$$

$$\mathbf{p}_{t|1:t-1} = \text{Softmax}(\mathbf{W}_p \mathbf{h}_t), \tag{2}$$

$$w_t \sim \mathbf{p}_{t|1:t-1}. \tag{3}$$

The procedure can be explained as follows. First, the latent state $\mathbf{h}_0$ is initialized to be zeros. At the t-th step, $\mathbf{h}_t$ is updated by an RNN cell g, which takes three inputs: the previous state $\mathbf{h}_{t-1}$, the word produced at the preceding step (represented by an embedded vector $\mathbf{x}_t$), and the visual feature $\mathbf{v}$. Here, the cell function g can take a simple form:

$$g(\mathbf{h}, \mathbf{x}, \mathbf{v}) = \tanh\left(\mathbf{W}_h \mathbf{h} + \mathbf{W}_x \mathbf{x} + \mathbf{W}_v \mathbf{v}\right). \tag{4}$$

More sophisticated cells, such as GRU [5] and LSTM [6], are also increasingly adopted in practice. To produce the word w_t, the latent state $\mathbf{h}_t$ will be transformed into a probability vector $\mathbf{p}_{t|1:t-1}$ via a fully-connected linear transform

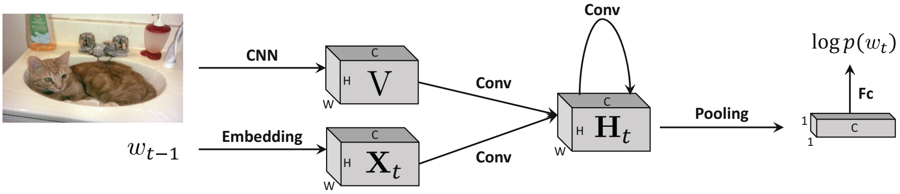


Fig. 1. The overall structure of the encoder-decoder framework with RNN-2DS. Given an image I, a CNN first turns it into a multi-channel feature map $\mathbf{V}$ that preserves high-level spatial structures. $\mathbf{V}$ will then be fed to an RNN-2DS, where the latent state $\mathbf{H}_t$ is also represented by multi-channel maps and the state transition is via convolution. At each step, the 2D states are transformed into a 1D vectors and then decoded into conditional probabilities of words.

$\mathbf{W}_p \mathbf{h}_t$ followed by a softmax function. Here, $\mathbf{p}_{t|1:t-1}$ can be considered as the probabilities of w_t conditioned on previous states.

Despite the differences in their architectures, all existing RNN-based captioning models represent latent states as *vectors* without explicitly preserving the spatial structures. In what follows, we will discuss the alternative choice that represents latent states as 2D multi-channel feature maps.

3.2 From 1D to 2D

From a technical standpoint, a natural way to maintain spatial structures in latent states is to formulate them as 2D maps and employ convolutions for state transitions, which we refer to as RNN-2DS.

Specifically, as shown in Fig. 1, the visual feature $\mathbf{V}$, the latent state $\mathbf{H}_t$, and the word embedding $\mathbf{X}_t$ are all represented as 3D tensors of size $C \times H \times W$. Such a tensor can be considered as a multi-channel map, which comprises C channels, each of size $H \times W$. Unlike the normal setting where the visual feature is derived from the activation of a fully-connected layer, $\mathbf{V}$ here is derived from the activation of a convolutional layer that preserves spatial structures. And $\mathbf{X}_t$ is the 2D word embedding for w_{t-1}, of size $C \times H \times W$. To reduce the number of parameters, we use a lookup table of smaller size $C_x \times H_x \times W_x$ to fetch the raw word embedding, which will be enlarged to $C \times H \times W$ by two convolutional layers[1]. With these representations, state updating can then be formulated using *convolutions*. For example, Eq. (4) can be converted into the following form:

$$\mathbf{H}_t = \text{relu}\left(\mathbf{K}_h \circledast \mathbf{H}_{t-1} + \mathbf{K}_x \circledast \mathbf{X}_t + \mathbf{K}_v \circledast \mathbf{V}\right). \quad (5)$$

Here, $\circledast$ denotes the convolution operator, and $\mathbf{K}_h$, $\mathbf{K}_x$, and $\mathbf{K}_v$ are convolution kernels of size $C \times C \times H_k \times W_k$. It is worth stressing that the modification

[1] In our experiments, the raw word embedding is of size 4 × 15 × 15, and is scaled up to match the size of latent states via two convolutional layers respectively with kernel sizes 32 × 4 × 5 × 5 and C × 32 × 5 × 5.

presented above is very flexible and can easily incorporate more sophisticated cells. For example, the original updating formulas of GRU are

$$\begin{aligned}
\mathbf{r}_t &= \sigma(\mathbf{W}_{rh}\mathbf{h}_{t-1} + \mathbf{W}_{rx}\mathbf{x}_t + \mathbf{W}_{rv}\mathbf{v}), \\
\mathbf{z}_t &= \sigma(\mathbf{W}_{zh}\mathbf{h}_{t-1} + \mathbf{W}_{zx}\mathbf{x}_t + \mathbf{W}_{zv}\mathbf{v}), \\
\tilde{\mathbf{h}}_t &= \tanh(\mathbf{r}_t \circ (\mathbf{W}_{hh}\mathbf{h}_{t-1}) + \mathbf{W}_{hx}\mathbf{x}_t + \mathbf{W}_{hv}\mathbf{v}), \\
\mathbf{h}_t &= \mathbf{z}_t \circ \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \circ \tilde{\mathbf{h}}_t,
\end{aligned} \tag{6}$$

where σ is the sigmoid function, and $\circ$ is the element-wise multiplication operator. In a similar way, we can convert them to the 2D form as

$$\begin{aligned}
\mathbf{R}_t &= \sigma(\mathbf{K}_{rh} \circledast \mathbf{H}_{t-1} + \mathbf{K}_{rx} \circledast \mathbf{X}_t + \mathbf{K}_{rv} \circledast \mathbf{V}), \\
\mathbf{Z}_t &= \sigma(\mathbf{K}_{zh} \circledast \mathbf{H}_{t-1} + \mathbf{K}_{zx} \circledast \mathbf{X}_t + \mathbf{K}_{zv} \circledast \mathbf{V}), \\
\tilde{\mathbf{H}}_t &= \mathrm{relu}(\mathbf{R}_t \circ (\mathbf{K}_{hh} \circledast \mathbf{H}_{t-1}) + \mathbf{K}_{hx} \circledast \mathbf{X}_t + \mathbf{K}_{hv} \circledast \mathbf{V}), \\
\mathbf{H}_t &= \mathbf{Z}_t \circ \mathbf{H}_{t-1} + (1 - \mathbf{Z}_t) \circ \tilde{\mathbf{H}}_t.
\end{aligned} \tag{7}$$

Given the latent states $\mathbf{H}_t$, the word w_t can be generated as follows. First, we compress $\mathbf{H}_t$ (of size $C \times H \times W$) into a C-dimensional vector $\mathbf{h}_t$ by mean pooling across spatial dimensions. Then, we transform $\mathbf{h}_t$ into a probability vector $\mathbf{p}_{t|1:t-1}$ and draw w_t therefrom, following Eqs. (2) and (3). Note that the pooling operation could be replaced with more sophisticated modules, such as an attention module, to summarize the information from all locations for word prediction. We choose the pooling operation as it adds zero extra parameters, which makes the comparison between 1D and 2D states fair.

Since this reformulation is generic, besides the encoder-decoder framework, it can be readily extended to other captioning models that adopt RNNs as the language module, *e.g.* Att2in [3] and Review Net [33].

4 Qualitative Studies on 2D States

Thanks to the preserved spatial locality, the use of 2D states makes the framework amenable to some qualitative analysis. Taking advantage of this, we present three studies in this section: (1) We manipulate the 2D states and investigate how it impacts the generated captions. The results of this study would corroborate the statement that 2D states help to preserve spatial structures. (2) Leveraging the spatial locality, we identify the associations between the activations of latent states and certain subregions of the input image. Based on the dynamic associations between state activations and the corresponding subregions, we can visually reveal the internal dynamics of the decoding process. (3) Through latent states we also interpret the connections between the visual and the linguistic domains.

4.1 State Manipulation

We study how the spatial structures of the 2D latent states influence the resultant captions by controlling the accessible parts of the latent states.

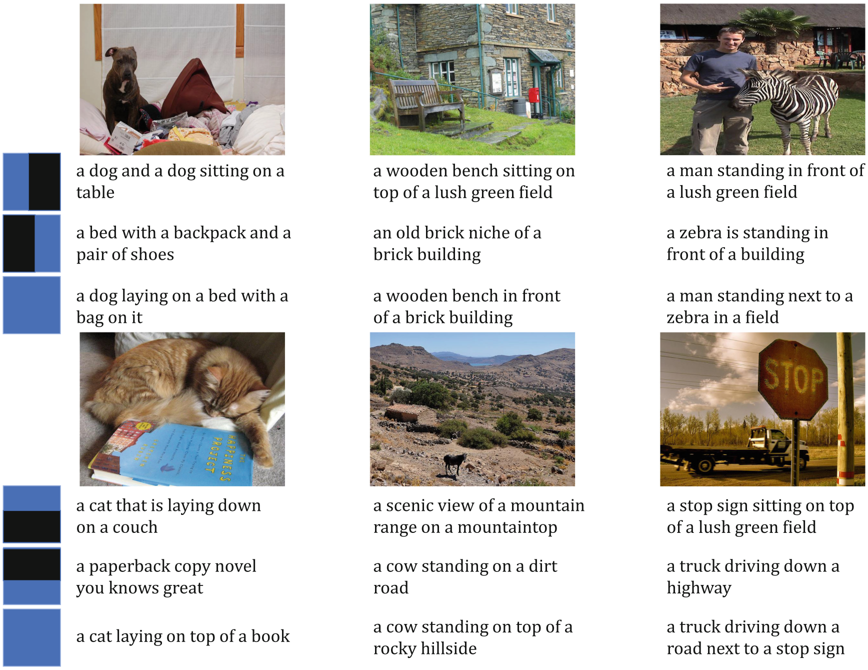

a dog and a dog sitting on a table
a bed with a backpack and a pair of shoes
a dog laying on a bed with a bag on it
a wooden bench sitting on top of a lush green field
an old brick niche of a brick building
a wooden bench in front of a brick building
a man standing in front of a lush green field
a zebra is standing in front of a building
a man standing next to a zebra in a field
a cat that is laying down on a couch
a paperback copy novel you knows great
a cat laying on top of a book
a scenic view of a mountain range on a mountaintop
a cow standing on a dirt road
a cow standing on top of a rocky hillside
a stop sign sitting on top of a lush green field
a truck driving down a highway
a truck driving down a road next to a stop sign

Fig. 2. This figure lists several images with generated captions relying on various parts of RNN-2DS's states. The accessible part is marked with blue color in each case. (Color figure online)

As discussed in Sect. 3.2, the prediction at t-th step is based on $\mathbf{h}_t$, which is pooled from $\mathbf{H}_t$ across H and W. In other words, $\mathbf{h}_t$ summarizes the information from the entire area of $\mathbf{H}_t$. In this experiment, we replace the original region $(1, 1, H, W)$ with a subregion between the corners (x_1, y_1) and (x_2, y_2) to get a modified summarizing vector $\mathbf{h}'_t$ as

$$\mathbf{h}'_t = \frac{1}{(y_2 - y_1 + 1)(x_2 - x_1 + 1)} \sum_{i=y_1}^{y_2} \sum_{j=x_1}^{x_2} \mathbf{H}_t|_{(i,j)}. \tag{8}$$

Here, $\mathbf{h}'_t$ only captures a subregion of the image, on which the probabilities for the word w_t is computed. We expect that this caption only partially reflects the visual semantics.

Figure 2 shows several images together with the captions generated using different subregions of the 2D states. Take the bottom-left image in Fig. 2 for an instance, when using only the upper half of the latent states, the decoder generates a caption focusing on the cat, which indeed appears in the upper half of the image. Similarly, using only the lower half of the latent states results in a caption that talks about the book located in the lower half of the image. In other words, depending on a specific subregion of the latent states, a decoder

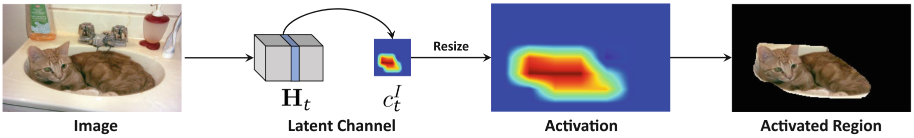


Fig. 3. This figure shows our procedure of finding the activated region of a latent channel at the t-th step.

with 2D states tends to generate a caption that conveys the visual content of the corresponding area in the input image. This observation suggests that the 2D latent states do preserve the spatial structures of the input image.

Manipulating latent states differs essentially from the passive data-driven attention module [2] commonly adopted in captioning models. It is a controllable operation, and does not require a specific module to achieve such functionality. With this operation, we can extend a captioning model with 2D states to allow *active* management of the focus, which, for example, can be used to generate multiple complementary sentences for an image. While the attention module can be considered as an automatic manipulation on latent states, the combination of 2D states and the attention mechanism worths exploring in the future work.

4.2 Revealing Decoding Dynamics

This study intends to analyze internal dynamics of the decoding process, *i.e.* how the latent states evolve in a series of decoding steps. We believe that it can help us better understand how a caption is generated based on the visual content. The spatial locality of the 2D states allows us to study this in an efficient and effective way.

We use *activated regions* to align the activations of the latent states at different decoding steps with the subregions in the input image. Specifically, we treat the channels of 2D states as the basic units in our study, which are 2D maps of activation values. Given a state channel c at the t-th decoding step, we resize it to the size of the input image I via bicubic interpolation. The pixel locations in I whose corresponding interpolated activations are above a certain threshold[2] are considered to be *activated*. The collection of all such pixel locations is referred to as the *activated region* for the state channel c at the t-th decoding step, as shown in Fig. 3.

With activated regions computed respectively at different decoding steps for one state channel, we may visually reveal the internal dynamics of the decoding process at that channel. Figure 4 shows several images and their generated captions, along with the activated regions of some channels following the decoding processes. These channels are selected as they are associated with nouns in the generated captions, which we will introduce in the next section. Via this study we found that (1) The activated regions of channels often capture salient visual

[2] See released code for more details.

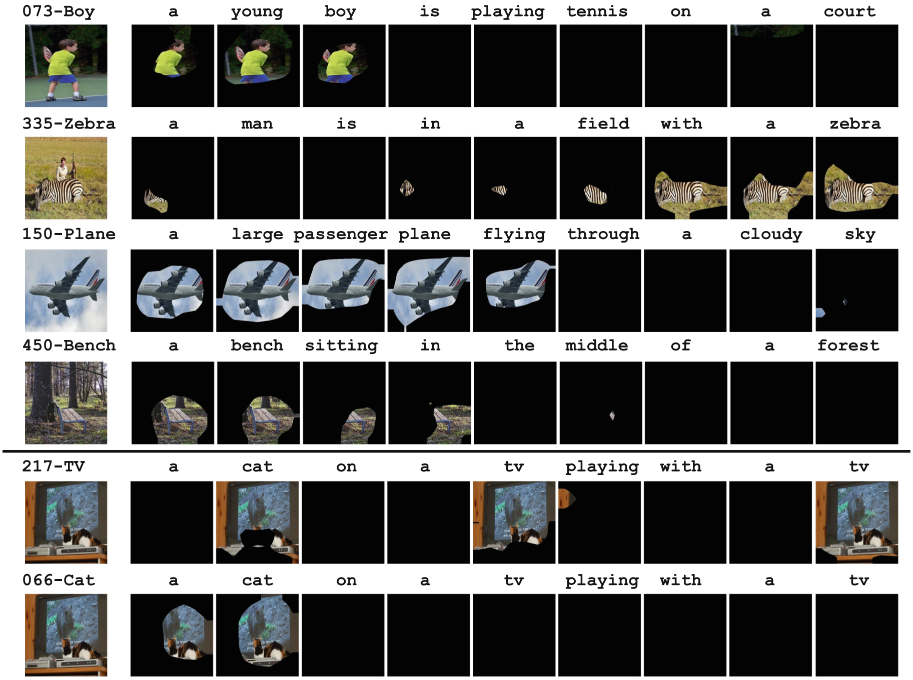


Fig. 4. This figure shows the changes of several channels, in terms of the activated regions, during the decoding processes. On the last two cases, changes of two channels in the same decoding process are shown and compared. (Best viewed in high resolution)

entities in the image, and also reflect the surrounding context occasionally. (2) During a decoding process, different channels have different dynamics. For a channel associated with a noun, the activated regions of its associated channel become significant as the decoding process approaches the point where the noun is produced, and the channel becomes deactivated afterwards.

The revealed dynamics can help us better understand the decoding process, which also point out some directions for future study. For instance, in Fig. 4, the visual semantics are distributed to different channels, and the decoder moves its focus from one channel to another. The mechanism that triggers such movements remains needed to be explored.

4.3 Connecting Visual and Linguistic Domains

Here we investigate how the visual domain is connected to the linguistic domain. As the latent states serve as pivots that connect both domains, we try to use the activations of the latent states to identify the detailed connections.

First, we find the associations between the latent states and the words. Similar to Sect. 4.2, we use state channels as the basic units here, so that we can use the activated regions which connect the latent states to the input image.

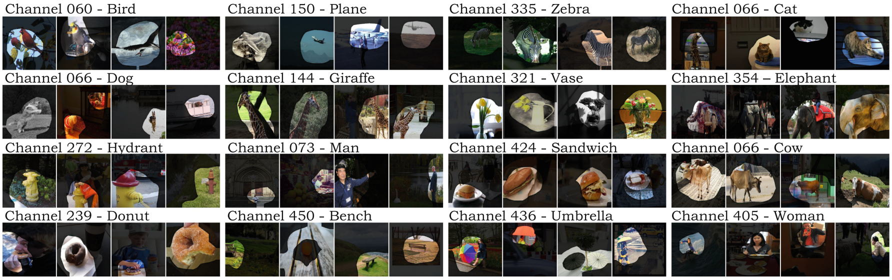


Fig. 5. Sample words and their associated channels in *RNN-2DS-(*$512, 7, 7$*)*. For each word, 5 activated regions of its associated channel on images that contain this word in the generated captions are shown. The activated regions are chosen at the steps where the words are produced. (Best viewed in high resolution)

In Sect. 4.2, we have observed that a channel associated with a certain word is likely to remain active until the word is produced, and its activation level will drop significantly afterwards thus preventing that word from being generated again. Hence, one way to judge whether a channel is associated with a word is to estimate the difference in its level of activations before and after the word is generated. The channel that yields *maximum difference* can be considered as the one associated with the word[3].

Words and Associated Channels. For each word in the vocabulary, we could find its associated channel as described above, and study the corresponding activated regions, as shown in Fig. 5. We found that (1) Only nouns have strong associations with the state channels, which is consistent with the fact that spatial locality is highly-related with the visual entities described as nouns. (2) Some channels have multiple associated nouns. For example, *Channel*-066 is associated with *"cat"*, *"dog"*, and *"cow"*. This is not surprising – since there are more nouns in the vocabulary than the number of channels, some nouns have to share channels. Here, it is worth noting that the nouns that share a channel tend to be visually relevant. This shows that the latent channels can capture meaningful visual structures. (3) Not all channels have associated words. Some channels may capture abstract notions instead of visual elements. The study of such channels is an interesting direction in the future.

Match of Words and Associated Channels. On top of the activated regions, we could also estimate the match between a word and its associated channel. Specifically, noticing the activated regions visually look like the attention maps in [34], we borrow the measurement of attention correctness from [34], to estimate the match. *Attention correctness* computes the similarity between a human-annotated segmentation mask of a word, and the activated region of its asso-

[3] See released code for more details.

Original	a red and red bird perched on a branch	a man getting ready to board a plane	a man standing in front of a fence with a bird	a vase filled with pink and yellow flowers
Deactivate word-associated channel	a red and green leaf filled with lots of fruit	a man standing next to a boarding gate	a man holding a baseball bat over his shoulder	a bouquet of red flowers sitting on a table

Fig. 6. This figure lists some images with generated captions before and after some word-associated channel being deactivated. The word that associates with the deactivated channel is marked in red. (Color figure online)

ciated channel, at the step the word is produced. The computation is done by summing up the normalized activations within that mask. On MSCOCO [14], we evaluated the attention correctness on 80 nouns that have human-annotated masks. As a result, the averaged attention correctness is 0.316. For reference, following the same setting except for replacing the activated regions with the attention maps, AdaptiveAttention [4], a state-of-the-art captioning model, got a result of 0.213.

Deactivation of Word-Associated Channels. We also verify the match of the found associations between the state channels and the words alternatively via an ablation study, where we compare the generated captions with and without the involvement of a certain channel. Specifically, on images that contain the target word w in the generated captions, we re-run the decoding process, in which we deactivate the associated channel of w by clipping its value to zero at all steps, then compare the generated captions with previous ones. As shown in Fig. 6, deactivating a word-associated channel leads to the miss of the corresponding words in the generated captions, even though the input still contains the visual semantics for those words. This ablation study corroborates the validity of our found associations.

5 Comparison on Captioning Performance

In this section, we compare the encoder-decoder framework with 1D states and 2D states. Specifically, we run our studies on MSCOCO [14] and Flickr30k [15], where we at first introduce the settings, followed by the results.

5.1 Settings

MSCOCO [14] contains 122, 585 images. We follow the splits in [35], using 112, 585 images for training, 5, 000 for validation, and the remaining 5, 000 for testing. Flickr30K [15] contains 31, 783 images in total, and we follow splits in

[35], which has $1,000$ images respectively for validation and testing, and the rest for training. In both datasets, each image comes with 5 ground-truth captions. To obtain a vocabulary, we turn words to lowercase and remove those with non-alphabet characters. Then we replace words that appear less than 6 times with a special token *UNK*, resulting in a vocabulary of size $9,487$ for MSCOCO, and $7,000$ for Flickr30k. Following the common convention [35], we truncated all ground-truth captions to have at most 18 words.

All captioning methods in our experiments are based on the encoder-decoder paradigm [1]. We use ResNet-152 [8] pretrained on ImageNet [9] as the encoder in all methods. In particular, we take the output of the layer `res5c` as the visual feature $\mathbf{V}$. We use the combination of the cell type and the state shape to refer to each type of the decoder. *e.g. LSTM-1DS-(L)* refers to a standard LSTM-based decoder with latent states of size L, and *GRU-2DS-(C, H, W)* refers to an RNN-2DS decoder with GRU cells as in Eq. (7), whose latent states are of size $C \times H \times W$. Moreover, all RNN-2DS models adopt a raw word-embedding of size $4 \times 15 \times 15$, except when a different size is explicitly specified. The convolution kernels $\mathbf{K}_h$, $\mathbf{K}_x$, and $\mathbf{K}_v$ share the same size $C \times C \times 3 \times 3$.

The focus of this paper is the representations of latent states. To ensure fair comparison, no additional modules including the attention module [2] are added to the methods. Moreover, no other training strategies are utilized, such as the scheduled sampling [36], except for the maximum likelihood objective, where we use the ADAM optimizer [37]. During training, we first fix the CNN encoder and optimize the decoder with learning rate 0.0004 in the first 20 epochs, and then jointly optimize both the encoder and decoder, until the performance on the validation set saturates.

For evaluation, we report the results using metrics including BLEU-4 (B4) [38], METEOR (MT) [39], ROUGE (RG) [40], CIDER (CD) [41], and SPICE (SP) [42].

5.2 Comparative Results

First, we compared *RNN-2DS* with *LSTM-1DS*. The former has 2D states with the simplest type of cells while the latter has 1D states with sophisticated LSTM cells. As the capacity of a model is closely related to the number of parameters, to ensure a fair comparison, each config of *RNN-2DS* is compared to an *LSTM-1DS* config *with a similar number of parameters*. In this way, the comparative results will signify the differences in the inherent expressive power of both formulations.

The resulting curves in terms of different metrics are shown in Fig. 7, in which we can see that *RNN-2DS* outperforms *LSTM-1DS* consistently, across different parameter sizes and under different metrics. These results show that *RNN-2DS*, with the states that preserve spatial locality, can capture both visual and linguistic information more efficiently.

We also compared different types of decoders with similar numbers of parameters, namely *RNN-1DS*, *GRU-1DS*, *LSTM-1DS*, *RNN-2DS*, *GRU-2DS*, and *LSTM-2DS*. Table 1 shows the results of these decoders on both datasets, from

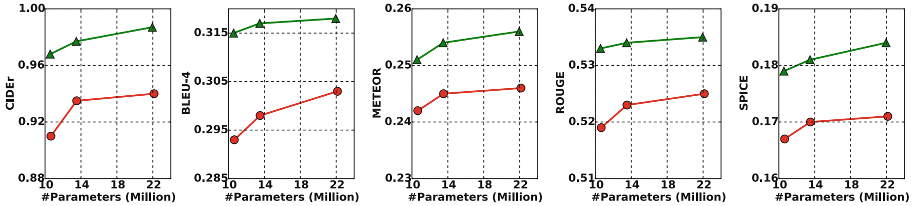


Fig. 7. The results, in terms of different metrics, obtained using RNN-2DS (green) and LSTM-1DS (red) on the MSCOCO offline test set with similar parameter sizes. Specifically, RNN-2DS of sizes 10.57M, 13.48M and 21.95M have compared to LSTM-1DS of sizes 10.65M, 13.52M and 22.14M. (Color figure online)

Table 1. The results obtained using different decoders on the offline and online test sets of MSCOCO, and the test set of Flickr30k, where METEOR (MT) [39] is omitted due to space limitation, and no SPICE (SP) [42] is reported by the online test set of MSCOCO.

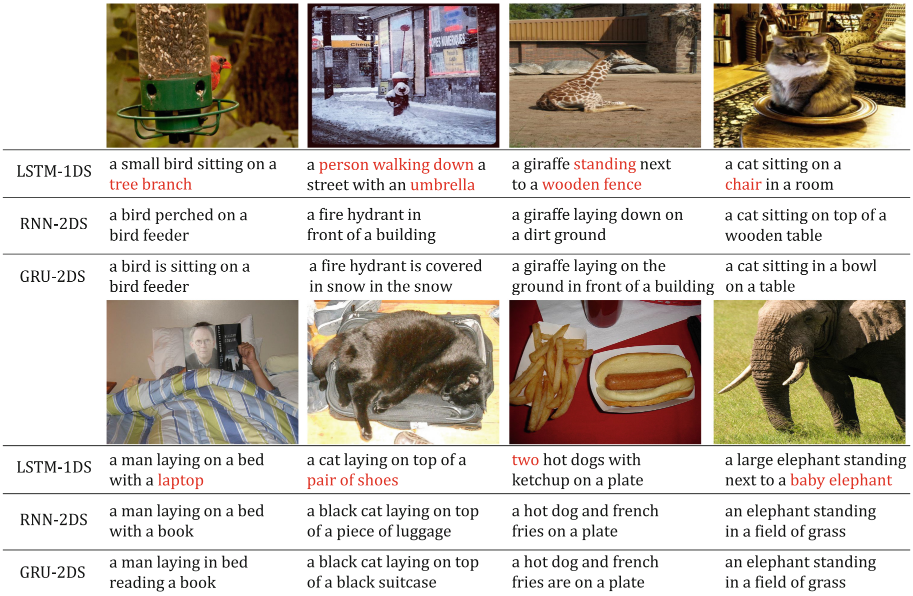
Model	#Param	COCO-offline				COCO-online			Flickr30k			
		CD	B4	RG	SP	CD	B4	RG	CD	B4	RG	SP
RNN-1DS-(595)	13.58M	0.914	0.293	0.520	0.168	0.868	0.286	0.515	0.353	0.195	0.427	0.117
GRU-1DS-(525)	13.53M	0.920	0.295	0.520	0.169	0.889	0.291	0.518	0.360	0.195	0.428	0.117
LSTM-1DS-(500)	13.52M	0.935	0.298	0.523	0.170	0.904	0.295	0.523	0.381	0.202	0.437	0.120
RNN-2DS-(256,7,7)	13.48M	0.977	0.317	0.534	0.181	0.930	0.305	0.527	0.420	0.217	0.442	0.125
GRU-2DS-(256,7,7)	17.02M	1.001	0.323	0.539	0.186	0.962	0.316	0.535	0.438	0.218	0.445	0.131
LSTM-2DS-(256,7,7)	18.79M	0.994	0.319	0.538	0.187	0.958	0.313	0.531	0.427	0.220	0.444	0.132

which we observe: (1) *RNN-2DS* outperforms *RNN-1DS*, *GRU-1DS*, and *LSTM-1DS*, indicating that embedding latent states in 2D forms is more effective. (2) *GRU-2DS*, which is also based on the proposed formulation but adds several gate functions, surpasses other decoders and yields the best result. This suggests that the techniques developed for conventional RNNs including gate functions and attention modules [2] are very likely to benefit RNNs with 2D states as well.

Figure 8 includes some qualitative samples, in which we can see the captions generated by *LSTM-1DS* rely heavily on the language priors, which sometimes contain the phrases that are not consistent with the visual content but appear frequently in training captions. On the contrary, the sentences from *RNN-2DS* and *GRU-2DS* are more relevant to the visual content.

5.3 Ablation Study

Table 2 compares the performances obtained with different design choices in *RNN-2DS*, including pooling methods, activation functions, and sizes of word embeddings, kernels and latent states The results show that mean pooling outperforms max pooling by a significant margin, indicating that information from all locations is significant. The table also shows the best combination of modeling choices for RNN-2DS: mean pooling, ReLU, the word embeddings of size 4 × 15 × 15, the kernel of size 3 × 3, and the latent states of size 256 × 7 × 7.



LSTM-1DS	a small bird sitting on a tree branch	a person walking down a street with an umbrella	a giraffe standing next to a wooden fence	a cat sitting on a chair in a room
RNN-2DS	a bird perched on a bird feeder	a fire hydrant in front of a building	a giraffe laying down on a dirt ground	a cat sitting on top of a wooden table
GRU-2DS	a bird is sitting on a bird feeder	a fire hydrant is covered in snow in the snow	a giraffe laying on the ground in front of a building	a cat sitting in a bowl on a table
LSTM-1DS	a man laying on a bed with a laptop	a cat laying on top of a pair of shoes	two hot dogs with ketchup on a plate	a large elephant standing next to a baby elephant
RNN-2DS	a man laying on a bed with a book	a black cat laying on top of a piece of luggage	a hot dog and french fries on a plate	an elephant standing in a field of grass
GRU-2DS	a man laying in bed reading a book	a black cat laying on top of a black suitcase	a hot dog and french fries are on a plate	an elephant standing in a field of grass

Fig. 8. This figure shows some qualitative samples of captions generated by different decoders, where words in red indicate they are inconsistent with the image. (Color figure online)

Table 2. The results obtained on the MSCOCO offline test set using RNN-2DS with different choices on pooling functions, activation functions, word-embeddings, kernels and latent states. Except for the first row, each row only lists the choice that is different from the first row. "-" means the same.

Pooling	Activation	Word-Embedding	Kernel	Latent-State	CD	B4	MT	RG	SP
Mean	ReLU	4 × 15 × 15	3 × 3	256 × 7 × 7	0.977	0.317	0.254	0.534	0.181
-	tanh	-	-	-	0.924	0.302	0.244	0.522	0.174
Max	-	-	-	-	0.850	0.279	0.233	0.507	0.166
-	-	1 × 15 × 15	-	-	0.965	0.313	0.251	0.532	0.180
-	-	7 × 15 × 15	-	-	0.951	0.309	0.250	0.529	0.179
-	-	-	1 × 1	-	0.927	0.298	0.247	0.522	0.177
-	-	-	5 × 5	-	0.951	0.308	0.250	0.529	0.177
-	-	-	-	256 × 5 × 5	0.934	0.300	0.245	0.523	0.173
-	-	-	-	256 × 11 × 11	0.927	0.300	0.246	0.523	0.176

6 Conclusions and Future Work

In this paper, we studied the impact of embedding latent states as 2D multi-channel feature maps in the context of image captioning. Compared to the standard practice that embeds latent states as 1D vectors, 2D states consistently achieve higher captioning performances across different settings. Such representations also preserve the spatial locality of the latent states, which helps reveal the internal dynamics of the decoding process, and interpret the connections between visual and linguistic domains. We plan to combine the decoder having 2D states with other modules commonly used in captioning community, including the attention module [2], for further exploration.

Acknowledgement. This work is partially supported by the Big Data Collaboration Research grant from SenseTime Group (CUHK Agreement No. TS1610626), the General Research Fund (GRF) of Hong Kong (No. 14236516).

References

1. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: a neural image caption generator. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3156–3164 (2015)
2. Xu, K., et al.: Show, attend and tell: Neural image caption generation with visual attention. In: ICML, vol. 14, pp. 77–81 (2015)
3. Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V.: Self-critical sequence training for image captioning. arXiv preprint arXiv:1612.00563 (2016)
4. Lu, J., Xiong, C., Parikh, D., Socher, R.: Knowing when to look: Adaptive attention via a visual sentinel for image captioning. arXiv preprint arXiv:1612.01887 (2016)
5. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
7. Graves, A.: Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850 (2013)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)
9. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. (IJCV) **115**(3), 211–252 (2015)
10. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: quantifying interpretability of deep visual representations. arXiv preprint arXiv:1704.05796 (2017)
11. Huang, Y., Wang, W., Wang, L.: Bidirectional recurrent convolutional networks for multi-frame super-resolution. In: Advances in Neural Information Processing Systems, pp. 235–243 (2015)
12. Romera-Paredes, B., Torr, P.H.S.: Recurrent instance segmentation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 312–329. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_19
13. Li, Z., Gavrilyuk, K., Gavves, E., Jain, M., Snoek, C.G.: Videolstm convolves, attends and flows for action recognition. Comput. Vis. Image Underst. **166**, 41–50 (2017)
14. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
15. Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: new similarity metrics for semantic inference over event descriptions. Trans. Assoc. Comput. Linguist. **2**, 67–78 (2014)

16. Kulkarni, G., et al.: Babytalk: understanding and generating simple image descriptions. IEEE Trans. Pattern Anal. Mach. Intell. **35**(12), 2891–2903 (2013)
17. Dai, B., Zhang, Y., Lin, D.: Detecting visual relationships with deep relational networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3298–3308. IEEE (2017)
18. Farhadi, A., et al.: Every picture tells a story: generating sentences from images. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 15–29. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_2
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
20. Dai, B., Fidler, S., Urtasun, R., Lin, D.: Towards diverse and natural image descriptions via a conditional GAN. In: Proceedings of the IEEE International Conference on Computer Vision (2017)
21. Dai, B., Lin, D.: Contrastive learning for image captioning. In: Advances in Neural Information Processing Systems, pp. 898–907 (2017)
22. Yao, T., Pan, Y., Li, Y., Qiu, Z., Mei, T.: Boosting image captioning with attributes. arXiv preprint arXiv:1611.01646 (2016)
23. Graves, A., Fernández, S., Schmidhuber, J.: Multi-dimensional recurrent neural networks. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D. (eds.) ICANN 2007. LNCS, vol. 4668, pp. 549–558. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74690-4_56
24. Zuo, Z., et al.: Convolutional recurrent neural networks: Learning spatial dependencies for image representation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 18–26 (2015)
25. Wu, Z., Lin, D., Tang, X.: Deep Markov random field for image modeling. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 295–312. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_18
26. Rui, C., Jing, Y., Rong-gui, H., Shu-guang, H.: A novel LSTM-RNN decoding algorithm in CAPTCHA recognition. In: 2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC), pp. 766–771. IEEE (2013)
27. Wang, C., Jiang, F., Yang, H.: A hybrid framework for text modeling with convolutional rnn. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2061–2069. ACM (2017)
28. Fu, X., Ch'ng, E., Aickelin, U., See, S.: CRNN: a joint neural network for redundancy detection. In: 2017 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 1–8. IEEE (2017
29. Keren, G., Schuller, B.: Convolutional RNN: an enhanced model for extracting features from sequential data. In: 2016 International Joint Conference on Neural Networks (IJCNN), pp. 3412–3419. IEEE (2016)
30. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.C.: Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: Advances in Neural Information Processing Systems, pp. 802–810 (2015)
31. Karpathy, A., Johnson, J., Fei-Fei, L.: Visualizing and understanding recurrent networks. arXiv preprint arXiv:1506.02078 (2015)
32. Ding, Y., Liu, Y., Luan, H., Sun, M.: Visualizing and understanding neural machine translation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 1150–1159 (2017)

33. Yang, Z., Yuan, Y., Wu, Y., Cohen, W.W., Salakhutdinov, R.R.: Review networks for caption generation. In: Advances in Neural Information Processing Systems, pp. 2361–2369 (2016)
34. Liu, C., Mao, J., Sha, F., Yuille, A.L.: Attention correctness in neural image captioning. In: AAAI, pp. 4176–4182 (2017)
35. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3128–3137 (2015)
36. Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N.: Scheduled sampling for sequence prediction with recurrent neural networks. In: Advances in Neural Information Processing Systems, pp. 1171–1179 (2015)
37. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
38. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318. Association For Computational Linguistics (2002)
39. Lavie, M.D.A.: Meteor universal: language specific translation evaluation for any target language. ACL 2014, p. 376 (2014)
40. Lin, C.Y.: Rouge: a package for automatic evaluation of summaries. In: Text Summarization Branches Out: Proceedings of the ACL 2004 Workshop, Barcelona, Spain, vol. 8 (2004)
41. Vedantam, R., Lawrence Zitnick, C., Parikh, D.: Cider: consensus-based image description evaluation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4566–4575 (2015)
42. Anderson, P., Fernando, B., Johnson, M., Gould, S.: SPICE: semantic propositional image caption evaluation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 382–398. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_24